

REMARKS

Claims 1, 3, 8, 9, 11, 12, 16, and 19 have been amended, and new claims 25-27 have been added. Accordingly, claims 1-27 remain pending.

The Examiner also rejected claims 1-7, 9-15, and 17-23 under 35 U.S.C. §103(a) as being unpatentable over Arunachalam (U.S. patent publication no. 2003/0069922) in view of McCullough (“Transparent Forwarding: First Steps”, pages 331-341). The Examiner has also rejected claims 8, 16, and 24 under 35 U.S.C. §103(a) as being unpatentable over Arunachalam in view of McCullough and further in view of Hawblitzel et al. (Implementing Multiple Domains in Java, pages 1-15). The Examiner’s rejections are respectfully traversed as follows.

Claim 1 is directed towards a method “for code from a first logical VM executing a remote method from a second logical VM.” The method of claim 1 comprises operations for “when code from a first logical VM is invoking a remote method of a second logical VM.” Claim 1 also requires wrapping each argument of the remote method when each argument is determined to be a remote object, wherein the first logical VM is associated with a first set of related code that are to be terminated together if misbehaving and the second logical VM is associated with a second set of related code that are to be terminated together if misbehaving, the first set of related code differing from the second set of related code.” In other words, the method operations are performed when code from a first logical VM are invoking a remote method of a second logical VM. The related code of each logical VM has been grouped together so as to be terminated together when such code is misbehaving.

Claim 1 also requires “copying each argument of the remote method when each argument is determined not a to be remote object.” In other words, each argument is wrapped when it is determined that each argument is a remote object. Otherwise, each argument is copied when it is determined that each argument is not a remote object. Claim 1 also requires “invoking the remote method using each wrapped or copied argument” and “wrapping the result of the invoked remote method when the result is determined to be a remote object.” Claim 1 also recites “copying the result of the remote method when the result is determined not to be a remote object.” That is, each result is wrapped when it is determined whether the result is a remote object. Otherwise, the result is simply copied. Claim 1 further requires “returning the wrapped or copied result only when an invocation thread associated with invoking the remote method of the second logical VM is not being terminated” and “throwing an exception on the remote method of the second logical VM when the invocation thread associated with invoking the remote method of the second logical VM is being terminated.” Independent claims 9 and 17 have similar limitations regarding the handling of remote arguments and results when a first logical VM is invoking a remote method of a second logical VM.

Embodiments of the present invention allow a first set of related code (referred to as a first logical VM) to invoke a remote method of another second set of related code (referred to as a second logical VM) in a way that allows termination of the second set of related code to not interfere with the first set of related code. For example, the invocation thread of the remote method of the second logical VM may have associated misbehaving threads or the invocation thread may itself be misbehaving. In this case, execution of the remote method is also terminated since it is a part of the second logical VM that is being terminated. In such a situation, the invocation process method is stopped and no results are returned from the remote method when the invocation thread that is invoking the remote method is being terminated. This prevents the terminating remote method from affecting the first logical VM's invoking process.

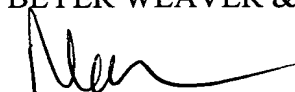
The primary reference Arunachalam is directed towards techniques for handling service transactions with multiple service providers. See Abstract. Although such transactions may involve passing objects between multiple applications on multiple service providers, such transactions do not include a method for code from a first logical VM to execute a remote method from a second logical VM. Also, Arunachalam does not provide a mechanism for minimizing the effect of a terminating logical VM on an invoking other logical VM. More specifically, Arunachalam fails to teach or suggest a method for invoking a remote object when code from a first logical VM is invoking a remote method of a second logical VM, wherein the first logical VM is associated with a first set of related code that are to be terminated together if misbehaving and the second logical VM is associated with a second set of related code that are to be terminated together if misbehaving, the first set of related code differing from the second set of related code. Arunachalam fails to teach or suggest the operations of "returning the wrapped or copied result only when an invocation thread associated with invoking the remote method of the second logical VM is not being terminated" and "throwing an exception on the remote method of the second logical VM when the invocation thread associated with invoking the remote method of the second logical VM is being terminated." Thus, Arunachalam would not reliably handle the termination of an invocation thread so as to prevent it from interfering with the invoking code, in the manner claimed. The secondary reference McCollough also fails to teach or suggest such limitations. Accordingly, it is respectfully submitted that claims 1, 9, and 17 are patentable over the cited art.

The Examiner's rejections of the dependent claims are also respectfully traversed. However, to expedite prosecution, all of these claims will not be argued separately. Claims 2-8, 10-16, and 18-24 each depend directly from independent claims 1, 9, and 17 and, therefore, are respectfully submitted to be patentable over cited art for at least the reasons set forth above with respect to claims 1, 9, and 17. Further, the dependent claims require additional elements that

when considered in context of the claimed inventions further patentably distinguish the invention from the cited art.

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP

A handwritten signature in black ink, appearing to read 'Mary Olynick', with a long horizontal flourish extending to the right.

Mary Olynick
Reg. 42,963

P.O. Box 778
Berkeley, CA 94704-0778
(510) 843-6200